



# Designing A High Quality Color Graphics Workstation

by Larke E. Reeber and Gunter Musolf

The desire to convert data into graphical forms that can be readily manipulated is making color graphics capability a requirement of many computer customers. This article discusses common methods of adding graphics to a desktop workstation and then discusses why

Convergent Technologies chose to use a specialized LSI processor dedicated to graphics display control, coupled with a general-purpose MOS  $\mu$ P, to implement the AWS Color Graphics Workstation.

## Character Set

An inexpensive and easy way to add graphics-like functions to an existing design is to enhance the character set by adding one or more ROMs or PROMs. Line drawing sets are created this way, as are some of the "graphics" on personal computers.

For simple forms generation or very simple bar charts this approach may be adequate. It requires little or no redesign of the alphanumeric display structure and is therefore quick to implement. But it is also inflexible since only predesignated graphic symbols are available. Typically no combining or overlaying of the symbols is possible either.

To increase flexibility this approach is often extended by using a RAM-based character set with an option to reload the character set or add more RAM for "user-de-

*Larke E. Reeber and Gunter Musolf are hardware engineers with Convergent Technologies, 2500 Augustine Dr., Santa Clara, CA 95051. (408) 727-8830.*



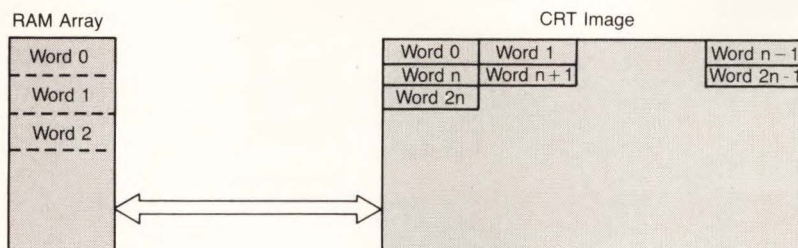


Figure 1: Correspondence between bit-mapped RAM array and image on CRT. There is a one-to-one correspondence between the words in a linear RAM array and the picture on the screen.

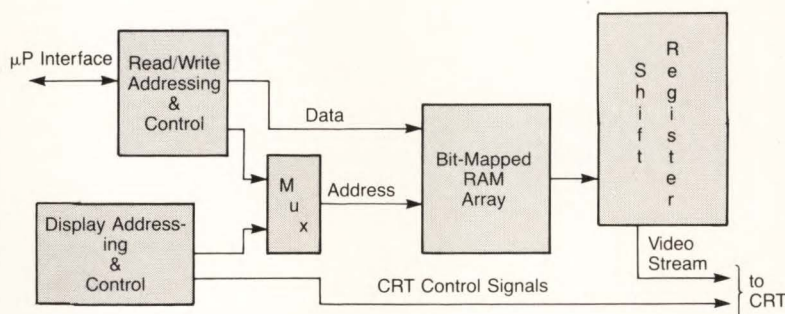


Figure 2: Bit mapped graphics video control. A typical bit-mapped graphics system shares a single address port into the RAM array.

finer" characters. This allows users the flexibility to design character cells that fit specific applications. For simple bar charts and pictures, where the necessary character cells are repetitive, this is usually sufficient. For more complex pictures, where the frequency of repetition of character cells is lower, the user may not have enough unique characters to complete the picture or graph. If there are not enough unique characters, the user must simplify the chart leaving out potentially valuable detail. To guarantee a sufficiently large number of unique characters requires a large RAM array with a short access time, thus eliminating the cost advantage of this structure.

Because of the problems associated with making enhanced character mode displays work for even moderately complex pictures and graphs, most raster-scan graphic implementations are bit-mapped. With a bit-mapped graphics sys-

tem, complex pictures and graphs with a lot of information to be displayed are no more difficult to show than simple ones with little information.

In a bit-mapped graphics system a RAM array that has a one-to-one correspondence with the visible image on the display is used to store the graphic image (Figure 1). This array must be controllable in two ways. First, the data in the RAM array must be read and sent to the CRT or other raster display. Second, the array must be modifiable so that storing the picture or graph

into it is not difficult. Some bit-mapped graphics systems are organized as dual-port memories with one address port for the display addressing and the other port for image generation. Most systems share a single address port with one function having priority (Figure 2).

### Controlling A Bit-Map Display

The design of a controller for a graphics bit map system requires understanding the overall system function. Graphics systems must be able to take objects and modify them and their characteristics on a display. Objects may be structured, that is, contain other objects, or be simple collections of vectors, arcs, and other primitive elements. These primitives must be manipulated to do translating (moving horizontally and vertically), rotating, scaling (changing the relative size on the screen), and clipping (deletion of invisible portions) of

*This approach to  
color graphics  
combines the 8086  
with the power of  
the 7220.*



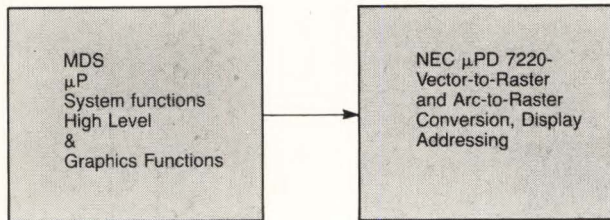


Figure 3: Partitioning of graphics functions with a 7220 adds a level of pipelining that increases throughput.

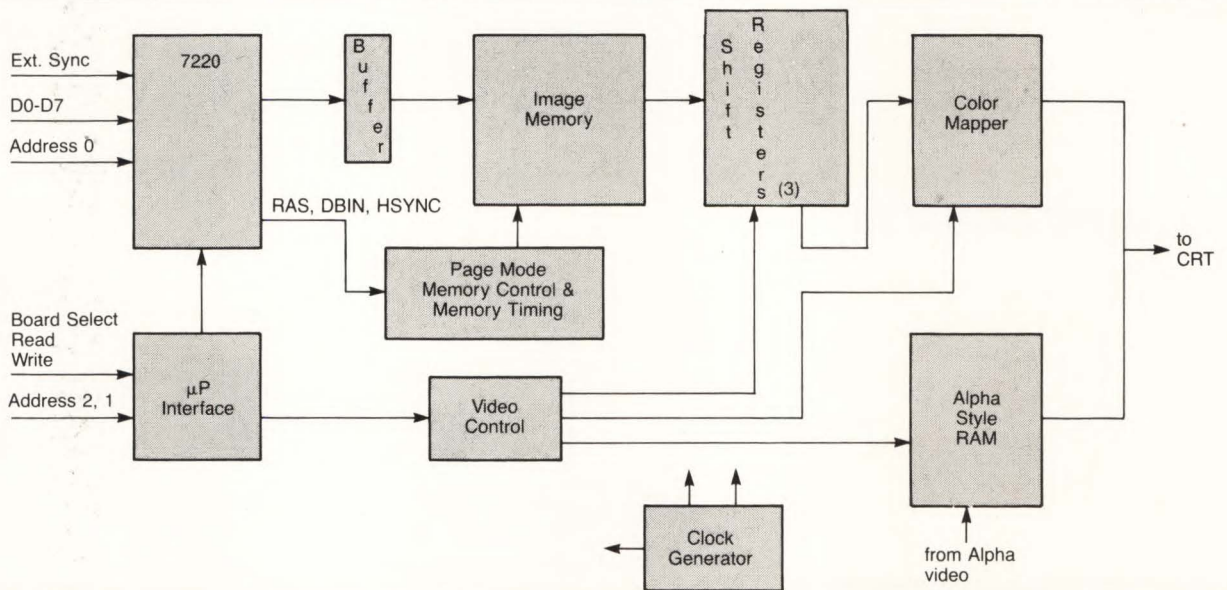


Figure 4: Block diagram of Convergent AWS graphics hardware architecture shows how the 7220 interfaces to the rest of the system.

the objects as the image is moved about on the screen. Once the vectors or arcs that need to be displayed are determined, then this information is converted into locations and data to be written. The process of converting vector information to raster RAM addresses is known as "vector-to-raster" conversion.

There are numerous ways to partition the sequence of tasks associated with converting data to pictures or charts. The most common way has all computations done by one processor. For large mainframe systems the host computer does all the computations and a "dumb terminal" holds the display. Personal computers take a similar approach on the opposite end of the scale. All system functions including all levels of graphic functions are done with one MOS μP.

Other partitionings include having a general-purpose MOS μP

handling system functions with a specialized finite state machine controlling a set of registers, counters, and adders which handle the vector-to-raster conversion. This adds a level of pipelining which increases throughput, but it also increases the amount of hardware needed. Substituting a MOS μP to handle the low level graphics manipulations in addition to the vector-to-raster conversion increases the flexibility and potential capabilities with little sacrifice in speed. Substituting a bipolar bit-slice processor for either processor increases throughput proportional to the increased processing speed, but bit-slice processor systems typically require excessive space and power. Having three levels of pipelining by separating the vector-to-raster conversions from the vector transformations from other system functions has a similar effect: an increase in speed for additional

board and power penalty.

In general, all these approaches suffer because they use general purpose ICs and processors, which are optimized for functions other than graphics. An LSI chip designed to handle graphics functions, the way USARTS handle data communications and CRT controller chips handle alphanumeric displays, is needed. NEC's μPD-7220 Graphics Display Controller is especially designed to handle graphics display address generation, vector-to-raster conversion, and arc-to-raster conversion. It does these functions at a fraction of the cost of the specially designed TTL versions. Although it does not perform vector transformations or associated matrix manipulations, it can simplify a design by combining many functions in one IC. It can also enhance performance by adding a layer of pipelining to the processor at a very reasonable cost (Figure 3).



Figure 6: Adjusting the 7220 clock timing to increase vector drawing speed.

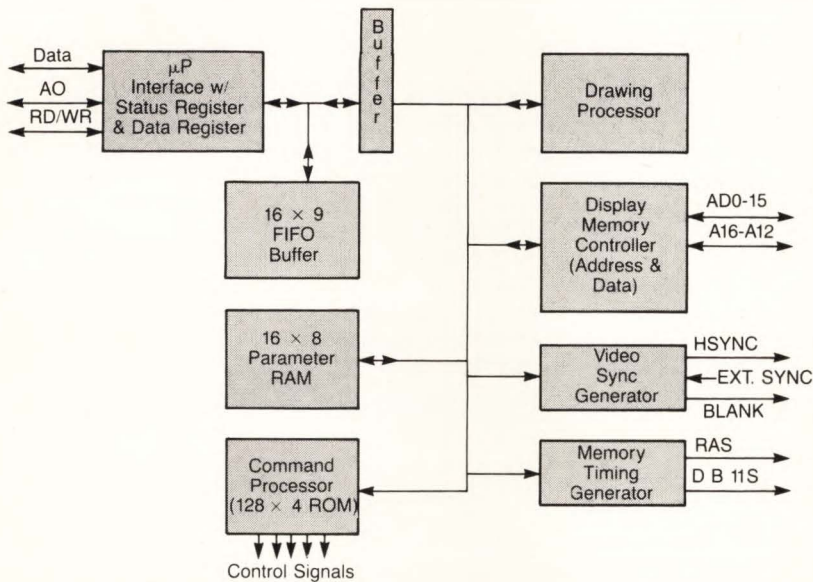
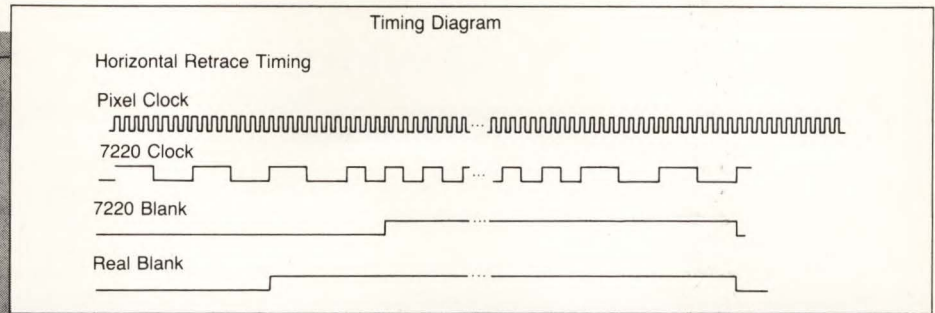


Figure 5: The 7220 internal architecture provides a FIFO to store commands and data, RAM to store parameters, and address and data generation for controlling the image memory.

## 7220 Based Graphics Design

The AWS Color Graphics Workstation consists of an 8 MHz 8086 running the CTOS Operating System as well as high level graphics software and a 7220 graphics controller handling low level graphics functions. The AWS Workstation also has a detachable keyboard, CRT display, communications ports, and optional mass storage. The video display is 319 lines by 432 pixels per line. An Intel 8275 CRT controller handles the alphanumeric display and supplies the CRT control signals.

In addition to the 7220, the AWS graphics board uses a page mode memory controller to get three bits/pixel using sixteen 64K dynamic RAMS. The three bits/pixel go through a color lookup table to provide a choice of 8 or of 64 colors. This information is then combined with the alphanumeric video

stream and sent to the display (Figure 4).

The 7220 (Figure 5) interfaces directly to a microcomputer bus and controls the bit-mapped image memory. By programming the designer can specify memory organization and screen timing. Drawing a vector or arc requires sending a series of commands and parameters which describe the item to be drawn. The 7220 then draws the vector during available memory cycles indicating data and address to be written.

### Interfacing to a $\mu$ C bus

One address bit indicates command or parameter when writing and indicates data or status when reading. An 8 bit bidirectional data bus contains the command, parameter, or data information. A simple board select, address, and read/write decoding provides the read and write signals for the 7220.

Interfacing to the display memory can be more complex. In the AWS Graphics Workstation implementation, the 7220 is operated in slave mode in order to synchronize it to the 8275 CRT Controller which generates the alphanumeric display. In slave mode the 7220 expects a minimum of eleven memory cycles during horizontal retrace, where a memory cycle is sixteen pixels long. Because the existing timing only has  $6\frac{1}{4}$  memory cycles, the clock speed is doubled during horizontal retrace except for the first half of the first cycle which is left unchanged (Figure 6). This gives twelve memory cycles during horizontal retrace and doubles the number of RMW cycles possible during blanking which increases the drawing speed. The 7220 also expects an even number of memory cycles during the visible portion of the horizontal scan. Because the system has 432 pixels/scanline



## A Low-Cost Color Monitor

Unlike other color systems which use external color monitors the AWS color monitor was developed to match the styling, software, and hardware of existing Convergent Workstations. This meant adherence to a relatively small packaging size, certain deflection speeds, and other electronic specifications.

The development started with the selection of a high resolution color picture tube. To reduce the circuitry requirements an in-line CRT was chosen. Use of the in-line CRT eliminates the need for convergence circuits, but requires a larger DC bias range for the three cathodes. Simplified circuits satisfy the DC bias range requirements and also provide RGB equalization of the tube characteristics.

Special modulation circuits are employed to provide electronic vertical and horizontal pin-cushion correction for the picture tube. A flyback transformer provides all voltages used in the color monitor except for one unregulated DC input voltage, which is supplied by an outside power supply.

The monitor is built to operate without a fan in a 60°C environment. A special outside heat sink is part of the integrated monitor.

Degaussing of the picture tube occurs automatically at AC power turn-on time and can also be manually activated by a push button switch, without interruption of input power.

The degaussing function operates on 110 VAC or 220 VAC at 50/60 Hz. The only other external control is an overall brightness adjustment.

The video amplifier input stage is designed to make the signal amplitudes controlling the CRT independent of input TTL levels. Gamma correction is inserted into each color channel to compensate for tube brightness nonlinearity and amplitude adjustment of the individual color channels is included in the video design.

High voltage and picture size regulation is provided by a number of feedback circuits encompassing the anode voltage and the DC input voltage to the flyback. Feedback circuits in the vertical amplifier make a short flyback time possible.

Fast rise and fall times in the video output stage with low overshoot at high voltage swings are accomplished by using shunt series inductor and emitter capacitor peaking. Arc protection for the circuitry is incorporated in the CRT socket for highest effectiveness.

Ease of adjustment in the video and deflection amplifiers is achieved by locating all controls at the edge of the PC boards. The circuit and PC board designs are adapted to accommodate the longer traces required without deterioration of performance. A special filter on the CRT screen reduces reflection from room light sources.

which, at 16 bits/word translates to 27 words/scanline, a special blanking circuit to blank the display one word early during each horizontal scan was added.

A method for accessing all three planes is needed, as the 7220 cannot manipulate more than one bit/pixel. The display memory is organized as a linear array of pixels with each plane contiguous. Page mode reads access the three planes for video display cycles using a counter for the two highest order address bits. For creating the picture, vectors are drawn three times, once at each plane.

When the video cycle is complete, the data is loaded into three parallel-to-serial shift registers where each pixel is shifted out to the color lookup table, an 8 × 6-bit RAM. The output of the lookup table is resynchronized on a pixel basis (**Figure 7**).

The alphanumeric display normally has three character attribute bits per character (halfbright, reverse, and underline in a monochrome system). An 8 × 8-bit color lookup table, separate from the graphics lookup table, uses the attributes to both select the color of the character and specify whether

the character is reversed or underlined. This information is resynchronized for each character (**Figure 7**). The character video stream then selects between the character information and the graphics information. If the character video stream is selected, then the character information is enabled and displayed. If

it is not selected, then the graphics pixels are displayed.

In addition, the alphanumeric displays and graphics displays can be selectively disabled. This allows the user the flexibility to display only text or only graphics or to combine the two with the text taking precedence. □

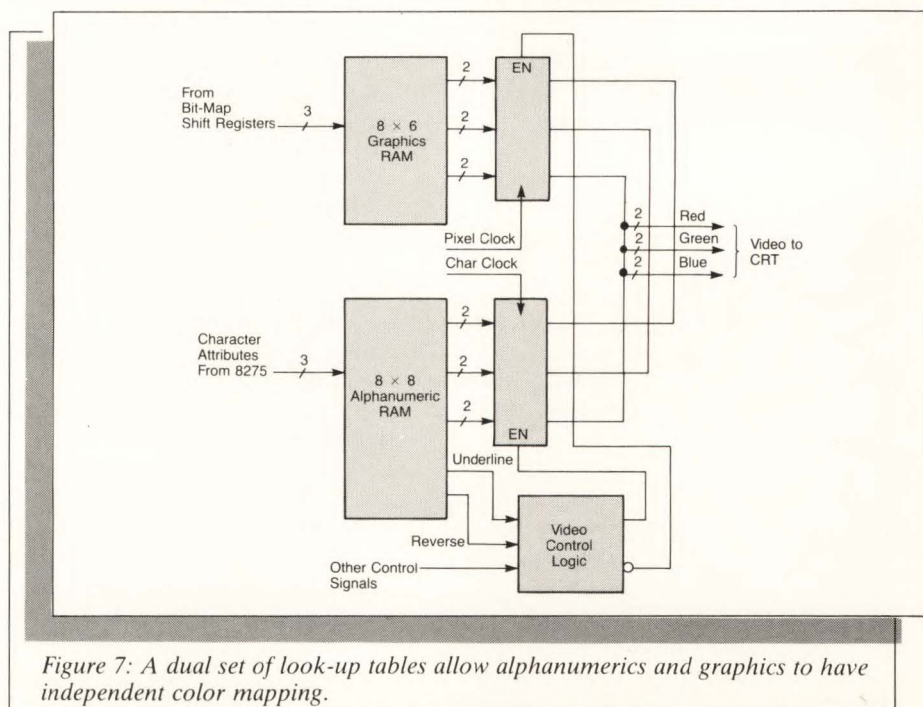


Figure 7: A dual set of look-up tables allow alphanumerics and graphics to have independent color mapping.